## **King Saud University**

College of computer and Information Sciences CSC 311 - Design and Analysis of Algorithms



Midterm Exam, Fall 2022

Wednesday Oct 5th, 2022

Exam time: 08:00-9:30 P.M.

Student's name:

## Problem 1 (11 p

(a) Give the following functions a number in order of increasing asymptotic growth rate. If two functions [3 pts] have the same asymptotic growth rate, give them the same number.

	Function	Rank
logn (ndn n3)	$3 \log n + \log \log \log (n^3)$	
	$\frac{3 \log n + \log \log \log (n^3)}{5 \lg n^5}$	
	$n^3$	2
	$\overline{\lg n}$	
	$n^3 + 6 \log n - 5$ $n^4  16^{\log n} + 3 n^2$	3
"n <sup>q</sup>	$n^{4} 16^{\log n} + 3n^{2}$	4 //
n	4 <sup>3n</sup>	5

(b) Using the definition of  $\theta$ , find g(n),  $C_1$ ,  $C_2$ , and  $n_0$  in the following:

[4 pts]

$$6n^{4} - 3n^{3} + n \log n \in \theta(g(n))$$

$$O(n^{4}) = 6n^{4} - 3n^{3} + n \log n \leq \log n^{4}$$

$$C_{1} = 10 \qquad \text{Not} = 1$$

$$\Sigma(n^{4}) = (n^{4} - 3n^{3} + n \log n), 5n^{4}$$

$$C_{2} = 5, noz = 4$$

$$\Theta(n^{4})$$
  $\begin{cases} C_{1} = 10 \\ C_{2} = 5 \end{cases}$   
 $\begin{cases} n_{0} \\ m_{0}x(n_{01}, n_{02}) = 4 \end{cases}$ 



جامعة الملك سعود كلية علوم الحاسب والمعلوصات 311 عال –

King Saud University

49e of computer and Information Sciences

SC 311 – Design and Analysis of Algorithms

[4 pts]

(c) Prove that log(n!) is O(n log n).

nl=n-1xn-2 xn-3--- complikity nl so ton is minemize the complixity to O(nlogh)

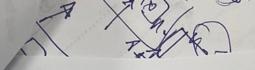
Problem 2 (7 points)



Consider the pseudo-code below:

a- Which problem does this algorithm solve?

It's sort the arrays by arrange the even numbers in the first them before any odd number (the thin)



**King Saud University** College of computer and Information Sciences CSC 311 - Design and Analysis of Algorithms



b- What is the time complexity (*Theta* (?)) of the following algorithm? Prove your answer (explain each step).

$$T(n) = 3n + 3$$
  $\Theta(n)$ 

$$T(n) = 3n+3$$
  $\Theta(n)$   
 $O(n) = \frac{1}{2}$ 

$$a(n) = \begin{cases} c = 2 \\ no = \frac{3}{3-2} = 3 \end{cases}$$

$$\Theta(n) = \begin{cases} c_1 = 6 \\ c_2 = 2 \end{cases}$$

$$s_1 = 3$$

كلية علوم السعاسب والمعلومات 311 عال -

**King Saud University College of computer and Information Sciences** 

CSC 311 - Design and Analysis of Algorithms

Consider the pseudo-code below, and assume the array A[..] is sorted in an increasing order:

```
S(A[..], key, imin, imax)
 if (imax < imin) t
      return KEY_NOT_FOUND; 1
 else
     imid = (imin+imax)/2;
     if (A[imid] > key)
      return S(A, key, imin, imid-1); T(1)
    else if (A[imid] < key)
                                                  T(n)=T(1/2)+4
      return S(A, key, imid+1, imax); -
    else -
     return imid; -
```

a- Which problem does this algorithm solve? Bainpy Search tofind the mid edlement

b- What is the design technique used in this solution. Explain your answer.

Devid and concer technique because we need be desired the element into two array edda time and then return the result



مسيدة علوم المحاسب والمعلومات 311 عال \_

71092

c- What is the time complexity of the following algorithm? Prove your answer.

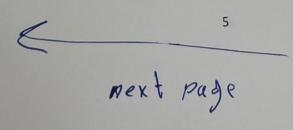
# Problem 4 (7 points)



Consider the following recurrence relation:

$$T(n)=4T(\frac{n}{4})+3n.$$

Solve this recurrence relation using recursive substitutions. Find g(n), where T(n) = O(g(n)).



### **King Saud University** College of computer and Information Sciences CSC 311 - Design and Analysis of Algorithms



$$T(n) = 4 \left[ 4 \left[ \frac{n}{4} \right] + 3 n \right]$$

$$T(\frac{n}{4}) = 4 \left[ 4 \left[ \frac{n}{16} \right] + 3 \frac{n}{4} \right] + 3 n = \frac{16}{16} \left[ \frac{n}{16} \right] + \frac{12}{4} \frac{n}{4} + 3 n$$

$$= \frac{64}{16} \left[ \frac{n}{64} + 64 \frac{n}{16} + \frac{12}{4} \frac{n}{4} + 3 n \right]$$

$$= \frac{64}{16} \left[ \frac{n}{64} + 64 \frac{n}{16} + \frac{12}{4} \frac{n}{4} + \frac{13}{4} n \right]$$

$$= \frac{64}{16} \left[ \frac{n}{64} + 64 \frac{n}{16} + \frac{12}{4} \frac{n}{4} + \frac{13}{4} n \right]$$

$$= \frac{64}{16} \left[ \frac{n}{64} + \frac{12}{4} \frac{n}{4} + \frac{13}{4} n \right]$$

$$= \frac{16}{16} \left[ \frac{n}{16} + \frac{13}{16} + \frac{13}{16} n \right]$$

$$= \frac{16}{16} \left[ \frac{n}{16} + \frac{13}{16} + \frac{13}{16} n \right]$$

$$= \frac{16}{16} \left[ \frac{n}{16} + \frac{13}{16} + \frac{13}{16} n \right]$$

$$= \frac{16}{16} \left[ \frac{n}{16} + \frac{13}{16} + \frac{13}{16} n \right]$$

$$= \frac{16}{16} \left[ \frac{n}{16} + \frac{13}{16} + \frac{13}{16} n \right]$$

$$= \frac{16}{16} \left[ \frac{n}{16} + \frac{13}{16} + \frac$$

Solve the following recurrence using the Master theorem by giving tight  $\theta$ -notation bounds. Justify your answers.

(a) 
$$T(n) = 3T(n/3) + 3 log^3(n)$$

3/0g(n) ( n10)3=n

 $\Theta(n)$ 





Problem 6 (10 points)

There are n parking spots numbered from 1 to n and you are told that there are k cars in the first k spots (one car in each spot), and all others. in each spot), and all other spots are empty. How to find the value of k?

(a) Suggest TWO algorithm design techniques and give a high-level description of the TWO algorithms to solve the problem (find k)?

1++; n

Deide and concern findk (A[1-n]) mid & n If (mid empty) 1 —
findk(A[1-..mid-1] T(1/2) return mid

T(n)= T(\frac{h}{2}) + 3
T(n)= 1+3logn



(b) Analyze the complexity of your TWO algorithms?

$$T(n) = T(\frac{n}{2}) + 3$$

$$T(n) \leqslant 1 + 3 \log n$$

$$O(10 + 3 \log n)$$

$$C = 4$$

$$n_0 = 1$$